A Locally Adaptive Penalty for Estimation of Functions with Varying Roughness.

Curtis B. Storlie, Howard D. Bondell, and Brian J. Reich

Date: January 28, 2010

Abstract

We propose a new regularization method called Loco-Spline for nonparametric function estimation. Loco-Spline uses a penalty which is data driven and locally adaptive. This allows for more flexible estimation of the function in regions of the domain where it has more curvature, without over fitting in regions that have little curvature. This methodology is also transferred into higher dimensions via the Smoothing Spline ANOVA framework. General conditions for optimal MSE rate of convergence are given and the Loco-Spline is shown to achieve this rate. In our simulation study, the Loco-Spline substantially outperforms the traditional smoothing spline and the locally adaptive kernel smoother. Code to fit Loco-Spline models is included with the Supplemental Materials for this article which are available online.

Keywords: Spatially Adaptive Smoothing, Nonparametric Regression, Regularization Method, Local Bandwidth, Smoothing Spline, L-Spline, SS-ANOVA.

Running title: Loco-Spline.

1 Introduction

Nonparametric Regression is a very useful approach to a large list of modern problems such as computer models, image data, environmental processes, to name a few. The nonparametric regression model is given by

$$y_i = f_0(x_i) + \varepsilon_i \;\; i = 1, \ldots, n,$$

where f_0 is an unknown regression function and ε_i are independent error terms. Smoothing splines are among the most popular methods for estimation of f_0 due to their good empirical performance and sound theoretical support (Cox 1983, Speckman 1985, Eubank 1999, van de Geer 2000, and many others). It is usually assumed without loss of generality that the domain of f_0 is $[0, 1]$. Let $f^{(m)}$ denote the m^{th} derivative of f. The smoothing spline estimate $\hat{f}$ is the unique minimizer of

$$\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_0^1 \left(f^{(m)}(x)\right)^2 dx \tag{1}$$

over all functions, f, in m^{th} order Sobolev space,

$$\mathcal{S}^m = \{f\colon\; f^{(j)} \text{ is absolutely continuous for } j = 1, \ldots, m-1 \text{ and } f^{(m)} \in L_2\}.$$

Notice that the penalty term on the right of (1) is an overall measure of the roughness of the function over the domain. The tuning parameter λ controls the trade-off in the resulting estimate between smoothness and fidelity to the data; large values of λ will result in smoother functions while smaller values of λ result in rougher functions but with better agreement to the data. Generally λ is chosen by generalized cross validation (GCV) (Craven & Wahba 1979), m-fold CV (Kohavi 1995), or related methods.

In many cases the underlying function changes more abruptly in some regions than in others. For example in structural engineering equations a beam may vibrate rapidly after a force is applied but the motion eventually becomes very smooth as it dies out. In situations like this the global penalty will cause the smoothing spline estimator to either over-smooth in some regions and/or under-smooth in others.

This paper considers the use of a locally varying smoothing parameter, $\lambda(x)$, which is a data driven function of x. This approach allows for more flexible estimation of the function in areas of the domain where the initial estimate has a large amount of curvature. This can be a large advantage when estimating functions that are very smooth in some areas, but have sharp peaks or abrupt changes in other parts of the domain.

The use of a local smoothing parameter is popular in kernel and local linear regression methods (Fan & Gijbels 1996). A major disadvantage to the use of kernel regression type methods is that these techniques do not translate well to estimation of functions with many predictors because of the well known "curse of dimensionality". Smoothing spline type optimizations on the other hand can work very well in the case of multidimensional predictors via the Smoothing Spline ANOVA (SS-ANOVA) framework (Wahba 1990, Lin 2000, Gu 2002). Hence there is much advantage to be gained from a locally adaptive smoothing spline type estimator.

There are also many approaches to surface fitting using spatially adaptive knot placement (basis function selection) with regression splines; see Friedman & Silverman (1989), Stone, Hansen, Kooperberg & Truong (1997), Luo & Wahba (1997), and Hansen & Kooperberg (2002). However, the properties of these estimators are difficult to study analytically since they are the result of an algorithm and not an explicit solution to an optimization problem. In addition, the stepwise nature of the algorithms can lead to instability of the final estimate. Lee (2004) is closer in spirit to the approach we take here. He calculates several smoothing spline estimates of varying smoothness, then chooses which of these estimates to use locally based on minimizing the local risk. This seems to work quite well at design points, but it is unclear how to define the estimator over the entire domain. When only a small to moderate number of observations are available or with multiple predictors this will become a significant problem.

Ruppert & Carroll (2000) use a penalization which is also similar in concept to our proposed method, but they restrict the estimate to a spline basis, making it more difficult to study convergence properties for a general space of functions. They impose a penalty on each of coefficients in the spline basis and allow the log of this penalty to vary as a linear spline. This requires the specification of M tuning parameters, $(\alpha_1^*, \dots, \alpha_M^*)$, one for each coefficient of the linear spline. This may be feasible for simple cases, but this approach suffers from the curse of dimensionality in higher dimensional predictor space. With only two predictors allowing for two way interaction would already require specification of M^2 different smoothing parameters. This will become computationally infeasible quite quickly as the number of predictors is increased.

Here we consider spatially adaptive estimators which are defined by the explicit function minimization problem,

$$\arg\min_{f \in \mathcal{S}^m} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \int_0^1 \lambda(x) \left(f^{(m)}(x) \right)^2 dx. \tag{2}$$

This formulation allows for the smoothing parameter to vary adaptively with x allowing for more/less penalty in regions of the domain where it is beneficial. Although the estimator in (2) is very flexible and intuitively appealing, it's implementation is very challenging without some simplifying assumptions on $\lambda(x)$. Pintore, Speckman & Holmes (2006) use a piecewise constant function for $\lambda(x)$ in (2). However, this form of $\lambda(x)$ has the same drawback as the penalty used in Ruppert & Carroll (2000). Namely, it requires specifying the number of knots, the knot locations, and the values of $\lambda(x)$ in-between the knot locations. This was accomplished by selecting one of several candidate knot location options and λ values between the knots via GCV. Unfortunately this leads to a smoothing method with a large number of smoothing parameters whose values need to be selected. Hence this approach also becomes cost prohibitive in higher dimensional predictor space. In addition, it may not be reasonable to assume that the smoothness of the function is very similar in-between knots, then changes abruptly at the knots. A continuously varying penalty would be more appropriate in most cases.

The contribution of this paper is the practical extension and the analysis of the theoretical properties of an idea proposed in Abramovich & Steinberg (1996). They proposed that the local smoothing parameter $\lambda(x)$ be chosen based on an initial estimate of the m^{th} derivative $f_0^{(m)}$. Unlike all previous attempts at locally adaptive spline smoothing, this framework requires only one smoothing parameter be chosen by cross validation.

Here we extend this concept, which we call *Loco-Spline*, to multiple predictors via SS-ANOVA. This is done with the same computational efficiency of the traditional smoothing spline procedure. Our numerical examples demonstrate a favorable performance of the Loco-Spline against several existing methods. In addition, we present general conditions for a local penalty function $\lambda(x)$ under which $\hat{f}$ converges at the optimal rate for nonparametric estimators. Thus, the added flexibility of Loco-Spline results in no loss of asymptotic optimality. To the best of our knowledge, this is the first result of its kind for any spatially adaptive spline type estimators.

The rest of the paper is laid out as follows. In Section 2 we present the Loco-Spline estimator in the univariate case. Section 2.2 then generalizes to higher dimensions via the SS-ANOVA framework. Theoretical properties for locally adaptive smoothing splines are given in Section 3. Section 4 discusses the computational considerations of Loco-Spline. Section 5 presents the results of applying the proposed methodology to several example problems and Section 6 concludes the paper.

2 Loco-Spline

We begin by introducing a special form of the Loco-Spline estimation problem for a univariate predictor, x, which has a motivating intuitive appeal. We then generalize this problem to the SS-ANOVA framework in Section 2.2.

2.1 Scatterplot Smoothing

Abramovich & Steinberg (1996) consider the solution to the minimization problem

$$\arg\min_{f\in\mathcal{S}^m}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \tau\int_0^1\left(\frac{f^{(m)}(x)}{\tilde{f}^{(m)}(x)}\right)^2 dx \tag{3}$$

over $f \in \mathcal{S}^m$ where $\tau > 0$ is a smoothing parameter and $\tilde{f}^{(m)}$ is an initial estimate of the m^{th} derivative of f_0. Notice that the contribution to the penalty in (3) is small in regions where the initial estimate has a lot of m^{th} order curvature (large m^{th} derivative). Hence the resulting estimator is able to have more curvature where it needs to without being over-penalized.

One potential disadvantage to the solution of (3) is that the resulting $\hat{f}$ is forced to have m^{th} order inflection points at exactly the same locations as in the initial estimate (i.e. $\hat{f}^{(m)}(x) = 0$ whenever $\tilde{f}^{(m)}(x) = 0$). This may not be ideal since we would like $\hat{f}$ to be somewhat robust to the choice of initial estimate. To overcome this issue, we now propose a general form for the Loco-Spline estimate. It is given by the minimizer over $f \in \mathcal{S}^m$ of the quantity

$$\arg\min_{f\in\mathcal{S}^m}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \tau\int_0^1\left(\frac{f^{(m)}(x)}{\left(\left|\tilde{f}^{(m)}(x)\right| + \delta\right)^{\gamma}}\right)^2 dx \tag{4}$$

for some constants $\delta \geq 0$ and $\gamma \geq 0$. The δ parameter allows for the release of the inflection restriction discussed above, while the γ parameter allows adjustment of the amount of weight placed in the initial estimate. The solution to (4) can be obtained in a fairly straight-forward manner using the reproducing kernel Hilbert space (RKHS) approach discussed in Wahba (1990) and Pintore et al. (2006). This solution is presented along with other computational details in Section 4.

There are many possible options for initial estimator $\tilde{f}^{(m)}$. We recommend taking the m^{th} derivative of the traditional smoothing spline estimate which penalizes the $(m+1)^{st}$ derivative. This is contrast to penalizing the m^{th} derivative as is done in Abramovich & Steinberg (1996). We propose the penalization on the $(m+1)^{st}$ derivative here because under certain conditions, this results in rate optimal estimation of $f_0^{(m)}$ when f_0 lies in $\mathcal{S}^{m+1}$ (Rice & Rosenblatt 1983). In our experience, this also seems to give much better (more stable and more precise) empirical results for the ultimate estimation of f_0 than penalization on the m^{th} derivative. On the other hand when $f_0 \in \mathcal{S}^m$ but $f^{(m+1)} \notin L_2$, then $\tilde{f}^{(m)}$ may not be rate optimal for $f_0^{(m)}$. However even in this case, the overall procedure still produces an asymptotically rate optimal estimator of f_0 (see Section 3) and still gives good empirical performance in our experience.

2.2 Extension to Multiple Predictors

With multiple predictors, other locally adaptive approaches either become computationally infeasible or suffer from the curse of dimensionality. Here we discuss the extension of Loco-Spline to multiple predictor variables, then demonstrate the ability of the Loco-Spline to avoid both of these issues. We will focus on the additive model for simplicity of presentation here. However, this framework described below easily extends to functions of any interaction order we might wish to consider in the SS-ANOVA decomposition.

To extend the problem to multiple predictors we need the following notation. Assume there are p predictor variables. Let x_j denote the value of the j^{th} predictor, $j = 1, \ldots, p$, and $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)$. It is assumed without loss of generality that $\boldsymbol{x} \in [0,1]^p$. Then let $x_{j,i}$ be the i^{th} observed value of the j^{th} predictor variable, $i = 1, \ldots, n$ and $\boldsymbol{x}_i = (x_{1,i}, x_{2,i}, \ldots, x_{p,i})$. Let $\mathcal{P}^m = \{f : \int_0^1 f^{(v)}(x)dx = 0,\ v = 0, \ldots, m-1\}$ which represent a certain type of periodic boundary constraints. Lastly, denote the space of additive m^{th} order Sobolev functions as $\mathcal{F} = \mathcal{S}_1^m \oplus \cdots \oplus \mathcal{S}_p^m = \{1\} \oplus \bar{\mathcal{S}}_1^m \oplus \cdots \oplus \bar{\mathcal{S}}_p^m$ where $\mathcal{S}_j^m$ is the m^{th} order Sobolev space corresponding to the j^{th} input variable, $\{1\}$ is the space of constant functions, and $\bar{\mathcal{S}}_j^m = \mathcal{S}_j^m \cap \mathcal{P}^1$. Hence $f \in \mathcal{F}$ implies

$f = b_0 + f_1 + \cdots + f_p$ for some $b_0 \in \Re$ and $f_j \in \bar{\mathcal{S}}_j^m$, $j = 1, \ldots, p$ which are called the functional components. Notice that the definition of $\bar{\mathcal{S}}_j^m$ implies that $\int_0^1 f_j(x)dx = 0$ for each j so that b_0 is identifiable.

The additive Loco-Spline estimator is now defined as

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{n}(y_i - b_0 - \sum_{i=1}^{p} f_i(x_{j,i}))^2 + \sum_{j=1}^{p} \tau_j \int_0^1 \left(\frac{f_j^{(m)}(x)}{\left(\left|\tilde{f}_j^{(m)}(x)\right| + \delta_j\right)^{\gamma_j}} \right)^2 dx_j \quad (5)$$

for initial estimates of $\tilde{f}_j^{(m)}$, $j = 1, \ldots, p$ and some user defined constants $\delta_j \geq 0$ and $\gamma_j \geq 0$ which play the same role as they did in the univariate case. Notice that the formulation in (5) requires specification of p smoothing parameters, τ_j's, one for each predictor. As in the typical additive model, these can be chosen via back-fitting as described in Section 4. Alternatively, one could use a common smoothing parameter $\tau_j = \tau$ for all j since the relative level of smoothness of the functional components is adjusted via the initial estimates. We do not assume a common smoothing parameter here however for two reasons: (i) The initial estimate, $\tilde{f}^{(m)}$ is best chosen by allowing for different smoothing parameters for each of the component curves so there would be little gain in computational efficiency anyway and (ii) in many cases the additive Loco-Spline performs better by allowing for different values for the γ_j's. If not all of the γ_j are equal, then the divisor is not on a comparable scale across components. Hence a separate tuning parameter would be necessary for each component.

3 Asymptotic Properties

Here we give some general conditions for which locally adaptive smoothing spline estimators in the additive model converge at the optimal rate for nonparametric regression estimators. As a corollary, Loco-Spline achieves this asymptotically optimal rate. To the best of our knowledge, this is the first result of its kind for a spatially adaptive smoothing spline estimator. Proofs of the following results are deferred to

Appendix A which is available as a supplemental document online.

Let the true regression function be in the space of additive m^{th} order Sobolev Space functions, $f_0 \in \mathcal{F} = \{1\} \oplus \bar{\mathcal{S}}_1^m \oplus \cdots \oplus \bar{\mathcal{S}}_p^m$. Denote $\|g\|_n^2 = 1/n \sum_{i=1}^n g(\boldsymbol{x}_i)^2$, the squared norm of the vector obtained by evaluating the function g at the design points. For two sequences a_n and b_n, we also use the notation $a_n \overset{p}{\sim} b_n$ to indicate $a_n/b_n = O_p(1)$ and $b_n/a_n = O_p(1)$.

Theorem 1. *Let $\hat{f}$ be given by the minimizer over $f = b_0 + f_1 + \cdots + f_p \in \mathcal{F}$ of the quantity $\sum_{i=1}^n (y_i - b_0 - \sum_{j=1}^p f_j(x_{j,i}))^2 + \sum_{j=1}^p \int_0^1 \lambda_{j,n}(x)(f_j^{(m)}(x))^2 dx$. Suppose that for each $x \in [0,1]$ the weight functions are such that $\max_{x\in[0,1]} \lambda_{j,n}(x) = O_p(n^{-2m/(2m+1)})$ and $\max_{x\in[0,1]} \lambda_{j,n}^{-1}(x) = O_p(n^{2m/(2m+1)})$ for each $j = 1, ..., p$. Then $||\hat{f} - f_0||_n^2 = O_p(n^{-2m/(2m+1)})$.*

Corollary 1. *Let $\hat{f}$ be given by the Loco-Spline estimate in (5) with $0 < \delta < \infty$ and $0 \leq \gamma < \infty$. Let $0 < M < \infty$ and set $\tilde{f}_j^{(m)}(x) = \operatorname{sign}\left(\bar{f}_j^{(m)}(x)\right) \min\left\{\left|\bar{f}_j^{(m)}(x)\right|, M\right\}$ where $\bar{f}_j^{(m)}$ is the estimate given by the traditional smoothing spline by penalizing the $(m+1)^{st}$ derivative. If also $\tau_{j,n} \overset{p}{\sim} n^{-2m/(2m+1)}$, then $||\hat{f} - f_0||_n^2 = O_p(n^{-2m/(2m+1)})$.*

Remark 1: Proofs of Theorem 1 and Corollary 1 can be found in the supplemental online document Appendix A. Corollary 1 implies that the Loco-Spline estimate obtains the asymptotically optimal rate for MSE convergence. Thus there is no loss in asymptotic rate of convergence when compared to the traditional additive smoothing spline (Lin 2000). However, there can be a substantial improvement in finite sample performance in some cases as seen in Section 5.

Remark 2: Note that the bound M on $\tilde{f}_j^{(m)}$ in Corollary 1 is introduced to satisfy the conditions on $\lambda_{j,n}(x)$ in Theorem 1. This ensures that for $g \in \mathcal{S}^m \cap \mathcal{P}^m$, then $g = 0$ if and only if $\int_0^1 [g^{(m)}(x)/(|\tilde{f}_j^{(m)}(x)| + \delta_j)^{\gamma_j}]^2 dx = 0$. Thus, this quantity can be thought of as a squared norm over the space $\mathcal{S}^m \cap \mathcal{P}^m$ (even in the limit as $n \to \infty$) just as the penalty for the traditional smoothing spline.

Remark 3: In Corollary 1 we calculate an initial estimate $\bar{f}$ by penalizing the $(m+1)^{st}$ derivative. If $f \in \mathcal{S}^{m+1}$, then the bound M would not be necessary because the resulting estimate of the m^{th} derivative, $\bar{f}^{(m)}$, would be bounded. So essentially, this M is only to guard against pathological cases resulting when $f \in \mathcal{S}^m$, but the m^{th} derivative of f is not bounded. Since we want the convergence rate to hold for *any* $f \in \mathcal{S}^m$, we need the extra condition involving M as a formality. In practice, M is a very large constant to simply protect against a poor estimate in the described pathological case. It is not really a truncation, in general, and in practice, M will almost never come into play. We set $M = 10^6$ for example, to generate the results for Section 5.

4 Computation

For ease of presentation, we first consider computation of the Loco-Spline estimate in the univariate case. This lays the groundwork for the computation in the general case which is discussed at the end of this section. The computation of the Loco-Spline solution is expedited by the use of reproducing kernel Hilbert space (RKHS) theory. We give a brief description of the concepts necessary for computation of the Loco-Spline solution. For a more in depth review of RKHS, see Wahba (1990) or Berlinet & Thomas-Agnan (2004).

4.1 RKHS solution

Recall that $\mathcal{P}^m = \{f : \int_0^1 f^{(v)}(x)dx = 0,\ v = 0, \dots, m-1\}$. Then $\mathcal{S}_0^m = \mathcal{S}^m \cap \mathcal{P}^m$ is the space of functions in m^{th} order Sobolev Space that satisfy the periodic boundary conditions. To calculate the general solution of the univariate Loco-Spline problem in (2), one simply needs the reproducing kernel (r.k.), $K_{m,\lambda}$, for the RKHS consisting of functions in $\mathcal{S}_0^m$ with inner product

$$\langle f, g \rangle_\lambda = \int_0^1 \lambda(x) f^{(m)}(x) g^{(m)}(x) dx.$$

The solution to (2) then has the form

$$\hat{f}(x) = \sum_{j=0}^{m-1} b_j B_j(x) + \sum_{i=1}^{n} c_i K_{m,\lambda}(x, x_i) \tag{6}$$

for some $\boldsymbol{b} = (b_0, \ldots, b_{m-1})'$ and $\boldsymbol{c} = (c_1, \ldots, c_n)'$, where B_j is the j^{th} Bernoulli polynomial. Hence, $\hat{f}$ can be obtained by simple matrix algebra after substituting (6) into (2); see Wahba (1990) for example.

The r.k., $K_{m,\lambda}$, for $\lambda(x) > 0$ and $\lambda(x)^{-1}$ square integrable is

$$K_{m,\lambda}(s,t) = \int_0^1 \lambda(u)^{-1} G_m(s,u) G_m(t,u) du, \tag{7}$$

where

$$G_m(s,t) = \frac{1}{m!} B_m(s) + \frac{(-1)^{m-1}}{m!} B_m(|s-t|) \left(\text{sign}(t-s)\right)^m \tag{8}$$

is the Green's function for the differential equation $f^{(m)}(x) = g(x)$ with the periodic boundary constraints described by $\mathcal{P}^m$ (Wahba 1995).

The r.k., $K_{m,\lambda}$, for the general form in (4) does not have a convenient closed form solution. However, one can numerically approximate the necessary integrals

$$\begin{aligned} K_{m,\lambda}(s,t) &= \frac{1}{\tau} \int_0^1 \left(\left| \tilde{f}^{(m)}(u) \right| + \delta \right)^{\gamma} G_m(s,u) G_m(t,u) du \qquad (9) \\ &\approx \frac{1}{N\tau} \sum_{k=1}^{N} \left(\left| \tilde{f}^{(m)}(u_k) \right| + \delta \right)^{\gamma} G_m(s,u_k) G_m(t,u_k) \qquad (10) \end{aligned}$$

for $u_k = (2k-1)/(2N)$. We have found that $N = 1000$ is sufficient for most cases. A Gauss quadrature could instead be used to numerically evaluate the integral, which generally requires fewer quadrature points. In our experience, the required computation time for the overall procedure is very similar either way, however. The Gram matrix, whose elements are the values of the kernel evaluated at the design points $K(x_i, x_j)$, $i, j = 1, \ldots, n$, is all that is needed to obtain the $\boldsymbol{b}$ and $\boldsymbol{c}$ of (6). To evaluate $\hat{f}$ at new x-values, we simply need to approximate $K_{m,\lambda}(x_{new}, x_i)$, $i = 1, \ldots, n$ in the same manner for the new x values.

The approximation in (10) requires that we know the derivative of $\tilde{f}$ at the quadrature points u_k. Fortunately, obtaining derivatives of $\tilde{f}$, when it is the standard $(m+1)^{st}$ order smoothing spline, is a fairly simple process. The functional form of the smoothing spline is given in Eq. (6). To obtain the derivative of the initial estimate $\tilde{f}$ then, we just need to take the derivative of the r.k. $K_{m,\lambda}(s,t)$ with respect to s. Derivatives of the r.k. for the standard $(m+1)^{st}$ order smoothing spline are fairly straight forward to evaluate analytically; see Wahba (1990) or Gu (2002) for the form of this r.k.

4.2 Tuning Parameter Selection

There are two free parameters in the Loco-Spline procedure, namely the traditional smoothing parameter, τ and the power given to the initial estimate of the m^{th} derivative, γ. Assume for now that we fix γ, then τ can be chosen via conventional means (GCV, m-fold CV, visually, etc.). Since the Loco-Spline procedure is not a linear smoother, it is perhaps best to use a method such as m-fold CV to choose τ. One could approximate the *df* of a nonlinear smoother as in Lin & Zhang (2006) and use GCV or similar measures. This would be somewhat faster computationally, but we have had better success with the 5-fold CV approach for this problem.

In our trials, we have found that the Loco-Spline estimate is not very sensitive to the exact value of δ. It suffices to use $\delta = 0.05 \max_{x\in[0,1]}\{\tilde{f}^{(m)}(x)\}$ to provide some freedom in the exact placement of inflection points. As it turns out though, it is helpful to more carefully consider the choice of γ. Although, this is much less crucial than the choice of τ it has been observed by the authors that certain functions tend to be better estimated with a larger value of γ. This is particularly true for functions that are very rough in isolated areas, but very smooth otherwise; see Section 5.1 for example. However, our experience also indicates that the choice of γ need not be all that precise. We have found that allowing the options of $\gamma = 1, 2, 4$ provides ample flexibility for most cases. Hence the algorithm used in the examples of Section 5 essentially fits a Loco-Spline estimator three times (once for each possible γ value), each time choosing τ via 5-fold CV. The final estimate uses the γ resulting in best 5-fold CV score. Thus, γ is technically a second tuning parameter in the manner it is used here. However, one can always fix $\gamma = 1$ to have a procedure with truly one tuning parameter which performs nearly as well in many cases.

4.3 Computation for the Additive Model

As in the univariate case, we set $\delta_j = 0.05 \max_{x \in [0,1]} \{\tilde{f}_j^{(m)}(x)\}$ to allow for some flexibility in the placement of inflection points in the final estimate. We will discuss the selection of the τ_j and γ_j, but we first consider the solution for fixed tuning parameters. In a similar fashion to the univariate problem, the solution to (5) has the form

$$\hat{f}(\boldsymbol{x}) = b_0 + \sum_{j=1}^{p} \sum_{k=1}^{m-1} b_{j,k} B_k(x_j) + \sum_{i=1}^{n} c_i K_{\boldsymbol{\tau},\tilde{f}}(\boldsymbol{x}, \boldsymbol{x}_i) \tag{11}$$

for some $\boldsymbol{b} = (b_0, \ldots, b_{m-1})'$ and $\boldsymbol{c} = (c_1, \ldots, c_n)'$, where recall that B_k is the k^{th} Bernoulli polynomial, and

$$K_{\boldsymbol{\tau},\tilde{f}}(\boldsymbol{s}, \boldsymbol{t}) = \sum_{j=1}^{p} \frac{1}{\tau_j} K_j(s_j, t_j)$$

where

$$K_j(s,t) = \int_0^1 |\tilde{f}_j^{(m)}(u) + \delta_j|^{\gamma_j} G_m(s,u) G_m(t,u) du.$$

Hence $\hat{f}$ can be obtained with simple linear algebra by substituting (11) into (5). The functions K_j for each $j = 1, \ldots, p$ must be evaluated at all pairwise combinations of the design points to obtain $\boldsymbol{b}$ and $\boldsymbol{c}$. This can be done as in (10).

For a given initial estimate, the algorithm to compute the additive Loco-Spline estimate including tuning parameter selection is given below. We discuss how to obtain the initial estimate immediately afterwards.

Algorithm 1.

```
1. Fix $\delta_j = 0.05 \max_{x \in [0,1]} \left| \tilde{f}_j^{(m)}(x) \right|$
2. Temporarily fix each $\tau_j = 1000$ and $\gamma_j = 0$ for all $j$.
3. for $j = 1, \ldots, p$
    (a) Keep all $\tau_k$ and $\gamma_k$ fixed unless $k = j$.
    (b) For $\gamma_j = \{1, 2, 4\}$, find the $\tau_j$ to minimize 5-fold CV score. This can be
        accomplished by solving (5) for each candidate value of $\log(\tau_j)$ on a grid
        for example.
    (c) Set $\gamma_j$ and $\tau_j$ at the values that minimized 5-fold CV score in the previous
        step (b).
4. Fix $\gamma_j$ at the value obtained in step 3 for the remainder of the algorithm.
5. Repeat step 3 (only adjusting $\tau_j$'s) a fixed number $K$ times or until some con-
   vergence criterion is satisfied.
```

Notice that the three levels of γ_j only get cross validated over for the first back-fitting iteration. This speeds up the overall algorithm considerably without much loss in performance from what we have observed. Finally, the full algorithm to compute Loco-Spline, including the initial estimate, is now given as

Algorithm 2.

```
1. Fit an initial additive model using the traditional smoothing spline penalizing
   the $(m+1)^{st}$ derivative. Specifically, use Algorithm 1 with $(m+1)^{st}$ derivative
   in (5) and $\gamma_j = 0$ for all $j$ fixed in step 3(b) for the entire algorithm to obtain
   $\tilde{f}$.
2. Use Algorithm 1 with the $\tilde{f}$ obtained in step 1 to obtain the Loco-Spline estimate.
```

5 Example Results

In this section we evaluate the performance of Loco-Spline on several simulated data sets and the benchmark motorcycle accident data used in Silverman (1985). We compare the results to those from several other competing methods. The methods included in these simulations are:

LOCO - the Loco-Spline procedure of (4) with tuning parameter selection via 5-fold CV as described in Section 4.2.

SAS(5) - the version of the spatially adaptive smoothing spline suggested in Pintore et al. (2006) which uses piecwise constant (with 5 bins since this had the best performance in their paper) for $\lambda(x)$.

TRAD - the traditional smoothing spline (TRAD) with tuning parameter chosen via 5-fold CV.

LOKERN - local kernel regression with plug-in local bandwidth as provided by the R package *lokern*. This procedure uses a second order kernel with a plug-in estimate of the asymptotically optimal local bandwidth.

MARS - Multivariate Adaptive Regression Splines (Friedman 1991) as provided by the R package *polymars*. This procedure uses regression splines with spatially adaptive knot placement.

Confidence intervals for $f(x)$ are obtained for these examples by means of the parametric "wild" bootstrap (Härdle 1990 and Davison & Hinkley 1997). It should be noted that confidence intervals for $f(x)$ could also be obtained by considering the posterior distribution of $f(x)$ from the equivalent Bayes model. Indeed, we can think of Loco-Spline as a Bayes estimate where the prior on f is a non-stationary Gaussian process with covariance given by $K_{m,\lambda}(s,t)$. This approach to calculating confidence intervals is shown to have desirable properties for the traditional smoothing spline (Nychka 1988). However, the Loco-Spline procedure makes heavy use of the data in estimating the "prior" covariance $K_{m,\lambda}(s,t)$. Hence this approach is likely to yield overly optimistic confidence intervals which makes the bootstrap approach more appropriate here.

5.1 Mexican Hat Function

The first test problem which we call the Mexican hat function is a linear function with a sharp Gaussian bump in the middle of the domain. Specifically the function is given by

$$f(x) = -1 + 1.5x + 0.2\phi_{0.02}(x - 0.5)$$

where $\phi_\sigma(x-\mu)$ is the $\mathcal{N}(\mu, \sigma^2)$ density evaluated at x. We generate a simple random sample of size n from $X_i \overset{iid}{\sim} \text{Unif}(0,1)$, $i = 1, \ldots, n$. We then generate $Y_i = f(X_i) + \varepsilon_i$, where $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, 0.25)$. We consider three scenarios for the sample size, $n = 100$, 250, and 500 to empirically observe the convergence of the methods.

Figure 1 displays the data along with the corresponding fits from Loco-Spline and the traditional smoothing spline for a typical realization with $n = 100$. Here we see that the Loco-Spline is able to both better capture the peak and stay smooth where the function is flat. In order for the traditional smoothing spline to estimate the peak reasonably well, the smoothing parameter needs to be small everywhere, hence allowing for the undesirable behavior of "chasing" data points in the areas where the true function is flat. Looking at the plot of the initial estimate of the second derivative (bottom left panel), we see that Loco-spline will be imposing far less penalty in the vicinity of the peak than in other regions. Hence the overall smoothing parameter need not be nearly as small relatively and no chasing of the data points occurs.

Bootstrap confidence intervals are plotted as bands in the upper right panel of the figure for the traditional smoothing spline and in the bottom right panel of the figure for Loco-Spline. Clearly the smoother and narrower confidence bands produced by Loco-Spline are preferable to those produced by TRAD.

In the top of Table 1 we can compare the performance on the Mexican hat example for these methods as sample size increases. The reported summary statistics are the average mean squared error (AMSE) and the percent best. The AMSE is the average of the MSE over 100 realizations at the respective sample sizes. Here we are using the definition of MSE which averages squared errors at the data points, i.e. $MSE = 1/n \sum_{i=1}^{n}(f(x_i) - \hat{f}(x_i))^2$. The percent best is the percentage of the 100 realizations that a given method had the smallest MSE among the competing methods.

In the Mexican hat section of the table it is quite evident that Loco-Spline is superior to the other approaches for all sample sizes on this example. Notice that, as sample size increases, the AMSE appears to be converging to zero at roughly the same rate for LOCO, TRAD and LOKERN, as predicted by their corresponding theoretical results. However, Loco-Spline maintains a smaller AMSE than the other methods at all sample sizes. In addition, Loco-Spline had the smallest MSE for roughly 80% of the realizations for each sample size.

5.2 Dampened Harmonic Motion

The next test problem is a dampened harmonic motion also known as the spring equation. Functions with this type of behavior are common to just about any structural engineering problem. The spring equation is given by

$$f(x) = a \exp\{-bx\} \cos(\omega x).$$

We have chosen the parameter values of $a = 1$, $b = 7.5$, $\omega = 10\pi$ to produce the data for this simulation. We again consider $X_i \overset{iid}{\sim} \text{Unif}(0, 1)$, $i = 1, \ldots, n$ with $Y_i = f(X_i) + \varepsilon_i$, but here $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, 0.05)$.

Figure 2 displays the data and the corresponding fits from Loco-Spline and the traditional smoothing spline for a typical realization with $n = 100$. Both the traditional smoothing spline and Loco-Spline capture the higher amplitude oscillation on the left third of the domain rather well. However, the traditional smoothing spline estimate is somewhat rough for $x > 0.4$ while Loco-Spline stays very smooth like the true function.

The second tier of Table 1 summarizes the performance on the dampened harmonic example for sample sizes $n = 100$, 250, and 500. In this example, there is not as much difference between the three methods LOCO, SAS(5), and TRAD. Still, LOCO and SAS(5) appear to be somewhat better than TRAD at all of the sample sizes. Even though the performances of LOCO and SAS(5) are similar, SAS(5) has smaller MSE in 75% of the realizations for $n = 250$ and $n = 500$.

5.3 Rapid Change Function

The rapid change function is defined as

$$f(x) = 1 - \frac{1}{1 + \exp\{-10(x - 0.2)\}} + \frac{0.8}{1 + \exp\{-75(x - 0.8)\}}$$

We once again consider $X_i \overset{iid}{\sim} \text{Unif}(0, 1)$ with $Y_i = f(X_i) + \varepsilon_i$ and $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, 0.05)$. Figure 3 displays the data and the corresponding fits from Loco-Spline and the traditional smoothing spline for a typical realization with $n = 100$. Notice how rough the smoothing spline is relative to the true function in regions away from the rapid change region ($x \approx 0.8$). Loco-Spline on the other hand is able to fit the true function just as well in the rapid change region while still producing a smooth and accurate estimate in the other regions.

Tier three of Table 1 summarizes the results of the simulations from this example. This example is more similar to the first example in that LOCO has a distinctly lower AMSE than the other methods at all of the sample sizes. LOCO has the smallest MSE in 68%, 80%, and 83% of the realizations at the three sample sizes $n = 100$, 250, and 500, respectively.

5.4 Computational Time

Here we consider the average CPU time required to fit each of the five methods considered in Sections 5.1-5.3. Figure 4 displays these CPU times versus sample size for each of the five methods. Notice that the computational cost to fit MARS and LOKERN is small relative to the other methods, but they also gave poorer performance in terms of estimation. The Loco-spline procedure takes substantially longer than the MARS, LOKERN, or TRAD methods to fit, but it also gave much better estimation accuracy in the simulations. The Loco-spline procedure also takes much less time to fit than the SAS(5) procedure of Pintore et al. (2006).

5.5 Motorcycle Crash Dataset

Here we take a look at a real data set that benefits from our local approach to smoothing. This data comes from a computer simulation of motorcycle accidents. The response is a series of measurements of head acceleration over time in a simulated motorcycle accident used to test crash helmets. This benchmark example dataset made popular by Silverman (1985) is available as *mcycle* in the R library MASS.

Figure 5 shows the estimated curves and confidence bands from TRAD and Loco-Spline respectively. Notice how Loco-Spline appears to have better agreement with the data at the three change points (13 sec, 22 sec, and 30 sec respectively), in that it captures the abrupt change without oversmoothing across the change points. On the other hand, Loco-Spline still maintains a very smooth nature between change points. This is particularly evident in the second half of the domain (30-60 sec). The TRAD estimate bounces around some in this region while Loco-Spline remains very smooth which seems to give a much more visually appealing fit to the data.

On performing a 10-fold CV of this data the CV scores for Loco-Spline, SAS(5), TRAD, LOKERN, and MARS are 535.9, 540.2, 544.3, 556.3, and 589.2, respectively. Hence Loco-Spline gives a much more visually appealing fit to this data set than TRAD and also has the lowest out of sample prediction error.

5.6 Additive Model Example

In this example we consider estimation of the following additive model,

$$f(x) = f_1(x_1) + f_2(x_2) + f_3(x_3) + f_4(x_4) + f_5(x_5)$$

where

$$\begin{aligned} f_1(x_1) &= 2x_1 \\ f_2(x_2) &= -1 + 1.5x + 0.2\phi_{0.02}(x - 0.5) \\ f_3(x_3) &= \exp\{-7.5x\}\cos(10\pi x) \\ f_4(x_4) &= 1 - \frac{1}{1 + \exp\{-10(x - 0.2)\}} + \frac{0.8}{1 + \exp\{-75(x - 0.8)\}} \\ f_5(x_5) &= 0 \end{aligned}$$

We generate a sample $\boldsymbol{X}_i = (X_{1,i}, \ldots, X_{5,i})$, $i = 1, \ldots, 100$ uniform on the unit cube, $[0, 1]^5$ and $Y_i = f(\boldsymbol{X}_i) + \varepsilon_i$, where $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, 0.125)$. Notice that f_1 and f_5 are very smooth functions where f_2, f_3, and f_4 are the functions with locally varying smoothness used as univariate examples in Sections 5.1 - 5.3. Figure 6 displays the data from a typical realization of this model along with the true components curves for the five predictor variables.

Figure 7 shows the true curves for the first four functional components along with the estimated curves from the traditional GAM model (Hastie & Tibshirani 1990) and the additive Loco-Spline model. The GAM estimate was produced using *Algorithm 1* with $m = 2$ and $\gamma_j = 0$ for all j fixed in step 3(b) for the entire procedure. Loco-Spline estimates were produced using *Algorithm 2* with $m = 2$. It can be seen here that both procedures do well to fit the functional components in regions where there is a lot of signal. However, the Loco-Spline estimate is much smoother in the regions of the domain where the true function is smooth. This is particularly true for f_2 and f_4. In addition, since Loco-Spline is capable of estimating the more complicated functions more precisely, it has a clearer picture of the remaining noise and can disregard noise variables like x_5. This can be seen in the bottom right panel of Figure 7 where Loco-spline correctly estimates f_5 to be nearly 0. GAM on the other hand picks up a substantial amount of spurious signal across x_5.

The last tier of Table 1 shows the summary of the MSE performance for Loco-Spline, GAM, and MARS on 100 realizations from this additive model. The other two methods (SAS(5) and LOKERN) were not used because they are not as readily extendable to the additive model due to speed and/or implementation concerns. It is quite clear that Loco-Spline is a much better procedure than the traditional GAM model for this example. Loco-Spline has AMSE=9.2, while GAM has AMSE=12.3 for the $n = 100$ case. MARS is again not competitive with the smoothing spline type procedures. Loco-Spline also had smaller MSE on 85 out of the 100 realizations. As sample size increases, the advantage of Loco-Spline is even more evident as it has universally better MSE in all of the realizations from $n = 250$ and $n = 500$.

6 Conclusions & Further Work

In this article, we have developed the Loco-Spline, a new regularization method which allows for a locally varying smoothness of the resulting estimate. We have demonstrated a favorable performance of this approach against several existing methods. The Loco-Spline method is then extended to additive models in multi-dimensional problems via SS-ANOVA, and compares favorably again to the traditional GAM model and MARS. It was also shown that even with the added flexibility to allow for better small sample performance, the Loco-Spline still achieves the asymptotically optimal rate of MSE convergence. R code to fit Loco-Spline models is included with the Supplemental Materials for this article which are available online.

There are certainly other questions and advancements still to be made with locally adaptive smoothing splines. For example many problems require treatment of two way (or higher) order interactions and/or variable selection. Also, a more detailed investigation of the first order term in the asymptotic MSE, perhaps by deriving the equivalent kernel would be useful. This could help make clear when advantage is gained over the traditional smoothing spline and give some insight into choice of tuning parameters.

Acknowledgments

The authors thank the National Science Foundation (Grant DMS-0705968) and Sandia National Laboratories (Grant SURP 22858) for partial support of this work. Lastly, the authors are grateful to the reviewers for their most constructive comments, most of which are incorporated in the current version of this article.

Supplemental Materials

Appendix: Proofs of the results given in Section 3. (pdf file)

R-code to fit Loco-spline models: R-code containing a function called `loco.spline` to fit the additive Loco-spline model described in this paper. The tar file also contains two files with sample code used to generate the results of Sections 5.1 and 5.6, respectively. (GNU zipped tar file)

References

Abramovich, F. & Steinberg, D. (1996), 'Improved inference in nonparametric regression using l_k-smoothing splines', *Journal of Statistical Planning and Inference* **49**, 327–341.

Berlinet, A. & Thomas-Agnan, C. (2004), *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Norwell, MA: Kluwer Academic Publishers.

Cox, D. (1983), 'Asymptotics for m-type smoothing splines', *Annals of Statistics* **11**(2), 530–551.

Craven, P. & Wahba, G. (1979), 'Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation', *Numerical Mathematics* **31**, 377–403.

Davison, A. & Hinkley, D. (1997), *Bootstrap Methods and their Application*, New York, NY: Cambridge University Press.

Eubank, R. (1999), *Nonparametric Regression and Spline Smoothing*, CRC Press.

Fan, J. & Gijbels, I. (1996), *Local Polynomial Models and its Applications*, Chapman & Hall, London.

Friedman, J. (1991), 'Multivariate adaptive regression splines (with discussion)', *Annals of Statistics* **19**, 1–141.

Friedman, J. & Silverman, B. (1989), 'Flexible parsimonious smoothing and additive modeling (with discussion)', *Technometrics* **31**, 3–39.

Gu, C. (2002), *Smoothing Spline ANOVA Models*, Springer-Verlag, New York, NY.

Hansen, M. & Kooperberg, C. (2002), 'Spline adaptation in extended linear models (with discussion)', *Statistical Science* **17**, 2–51.

Härdle, W. (1990), *Applied Nonparametric Regression*, New York, NY: Cambridge University Press.

Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman & Hall/CRC.

Kohavi, R. (1995), 'A study of cross-validation and bootstrap for accuracy estimation and model selection', *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* **2**(12), 1137–1143.

Lee, T. (2004), 'Improved smoothing spline regression by combining estimates of different smoothness', *Statistics & Probability Letters* **67**, 133–140.

Lin, Y. (2000), 'Tensor product space anova models', *Annals of Statistics* **28**(3), 734–755.

Lin, Y. & Zhang, H. (2006), 'Component selection and smoothing in smoothing spline analysis of variance models', *Annals of Statistics* **34**(5), 2272–2297.

Luo, Z. & Wahba, G. (1997), 'Hybrid adaptive splines', *Journal of the American Statistical Association* **92**(437), 107–116.

Nychka, D. (1988), 'Bayesian confidence intervals for smoothing splines', *Journal of the American Statistical Association* **83**, 1134–1143.

Pintore, A., Speckman, P. & Holmes, C. (2006), 'Spatially adaptive smoothing splines', *Biometrika* **93**(1), 113–125.

Rice, J. & Rosenblatt, M. (1983), 'Smoothing splines: Regression, derivatives and deconvolution', *Annals of Statistics* **11**(1), 141–156.

Ruppert, D. & Carroll, R. (2000), 'Spatially adaptive penalties for spline fitting', *New Zealand Journal of Statistics* **42**, 205–223.

Silverman, B. (1985), ‘Some aspects of the spline smoothing approach to non-parametric curve fitting’, *Journal of the Royal Statistical Society: Series B* **47**, 1–52.

Speckman, P. (1985), ‘Spline smoothing and optimal rates of convergence in nonparametric regression-models’, *Annals of Statistics* **13**(3), 970–983.

Stone, C., Hansen, M., Kooperberg, C. & Truong, Y. (1997), ‘1994 wald memorial lectures - polynomial splines and their tensor products in extended linear modeling’, *Annals of Statistics* **25**(4), 1371–1425.

van de Geer, S. (2000), *Empirical Processes in M-Estimation*, Cambridge University Press.

Wahba, G. (1990), *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics.

Wahba, G. (1995), ‘Wavelet shrinkage: Asymptopia? - discussion’, *Journal of the Royal Statistical Society: Series B* **57**(2), 360–361.

	$n = 100$		$n = 250$		$n = 500$	
	AMSE	% Best	AMSE	% Best	AMSE	% Best
	Mexican Hat					
LOCO	11.16 (0.66)	78.0	4.74 (0.28)	83.0	2.44 (0.10)	79.0
SAS(5)	15.38 (0.60)	14.0	6.43 (0.22)	15.0	3.23 (0.10)	21.0
TRAD	20.88 (0.53)	1.0	9.46 (0.19)	2.0	5.25 (0.10)	0.0
LOKERN	19.25 (0.53)	5.0	10.11 (0.21)	0.0	5.63 (0.12)	0.0
MARS	60.01 (3.88)	2.0	61.56 (3.92)	0.0	52.24 (3.30)	0.0
	Dampened Harmonic					
LOCO	0.57 (0.02)	31.0	0.23 (0.01)	22.0	0.13 (0.00)	23.0
SAS(5)	0.54 (0.03)	58.0	0.20 (0.01)	75.0	0.11 (0.00)	75.0
TRAD	0.60 (0.02)	8.0	0.26 (0.01)	3.0	0.15 (0.00)	2.0
LOKERN	0.78 (0.03)	3.0	0.34 (0.01)	0.0	0.19 (0.00)	0.0
MARS	1.88 (0.21)	0.0	1.51 (0.17)	0.0	0.92 (0.07)	0.0
	Rapid Change					
LOCO	0.42 (0.02)	68.0	0.18 (0.01)	80.0	0.10 (0.00)	83.0
SAS(5)	0.53 (0.02)	14.0	0.23 (0.01)	12.0	0.12 (0.00)	12.0
TRAD	0.54 (0.02)	3.0	0.26 (0.01)	1.0	0.14 (0.00)	1.0
LOKERN	0.59 (0.02)	4.0	0.29 (0.01)	1.0	0.16 (0.00)	0.0
MARS	0.71 (0.03)	9.0	0.39 (0.02)	5.0	0.25 (0.01)	4.0
	Additive Function					
LOCO	9.20 (0.48)	85.0	3.72 (0.09)	99.0	2.17 (0.09)	97.6
GAM	12.26 (0.29)	15.0	5.71 (0.10)	1.0	3.09 (0.08)	2.4
MARS	86.49 (4.61)	0.0	85.64 (5.53)	0.0	72.83 (8.97)	0.0

Table 1: Results of 100 Realizations from each of the examples models: Mexihat, Dampened Harmonic, Rapid Change, and Additive Function. AMSE is the mean square error averaged over the 100 realizations; standard error in parentheses. The percentage of the realizations that a particular method had the smallest MSE among the other methods is given as % Best.

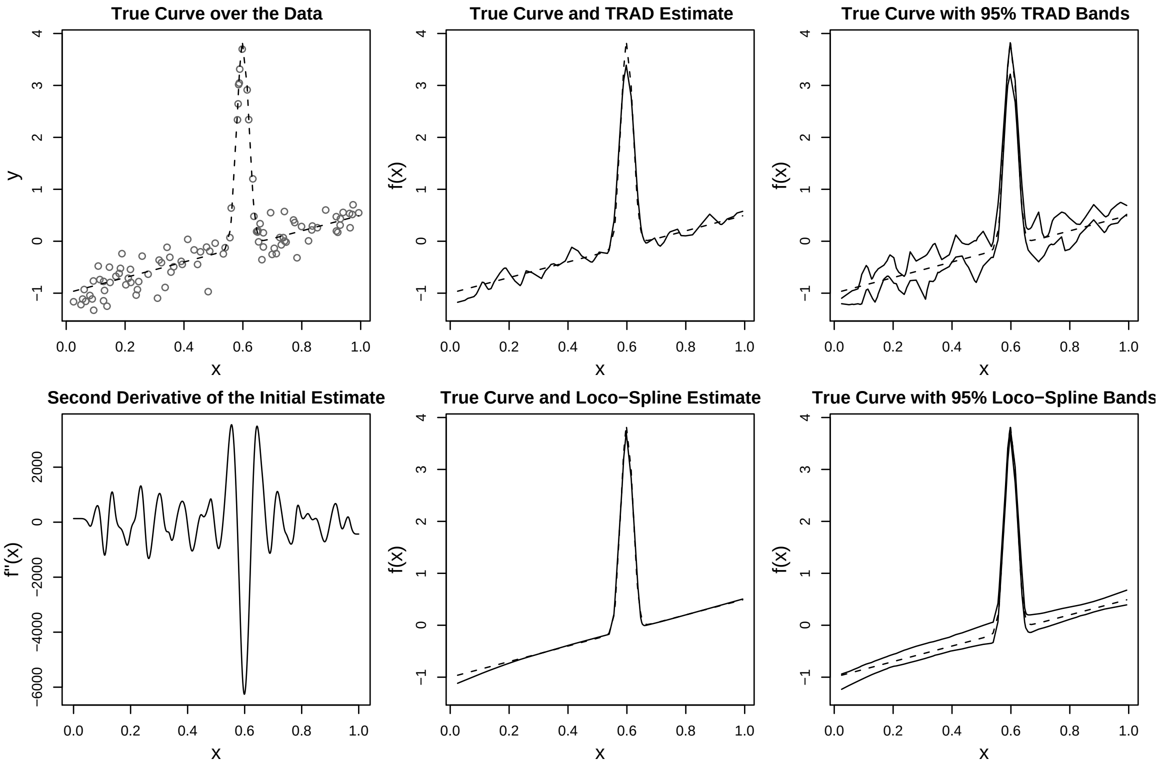

Figure 1: *Upper left*: Data generated from the Mexican hat function with $n = 100$ along with the true function. *Upper middle*: The traditional smoothing spline estimate (solid) with the true function (dashed). *Upper right*: 95% bootstrap confidence bands obtained from the traditional smoothing spline (solid) with the true function (dashed). *Lower left*: The curvature of the initial estimate (obtained using $m = 3$) used to weight the smoothing parameter. *Lower middle*: The Loco-spline estimate (solid) with the true function (dashed). *Lower right*: 95% bootstrap confidence bands obtained from the Loco-Spline procedure (solid) with the true function (dashed).

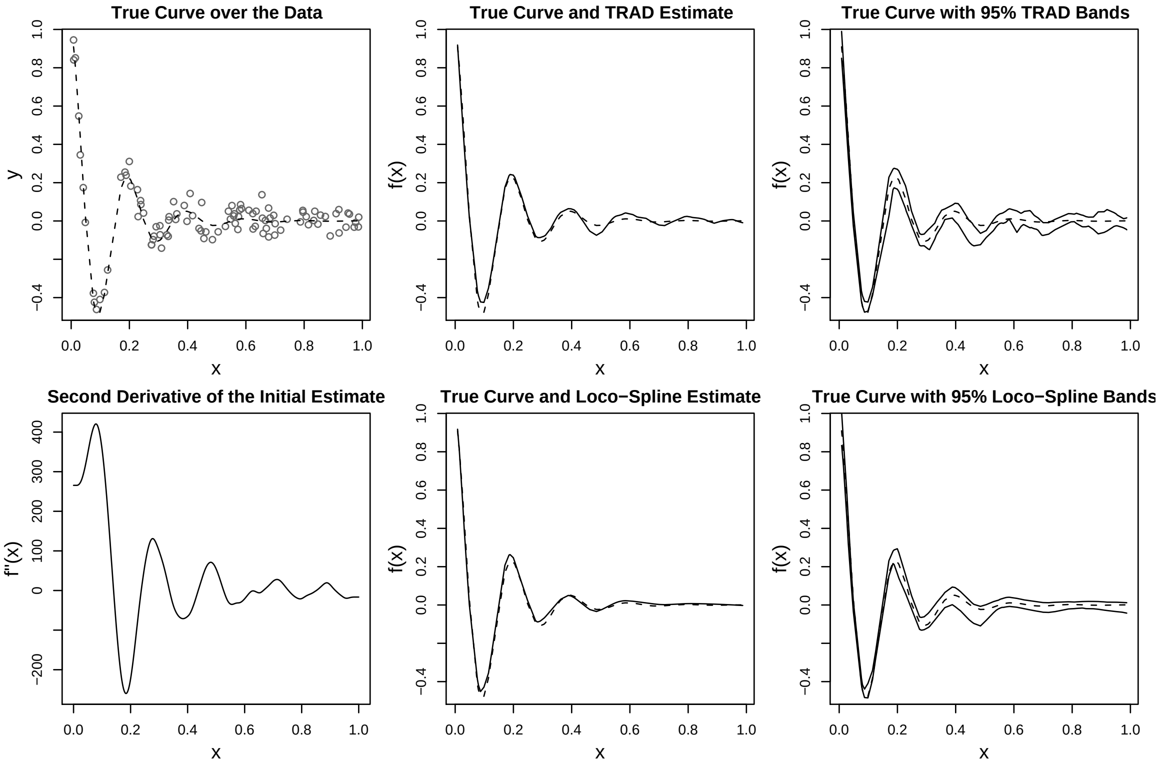

Figure 2: *Upper left*: Data generated from the dampened harmonic function with $n = 100$ along with the true function. *Upper middle*: The traditional smoothing spline estimate (solid) with the true function (dashed). *Upper right*: 95% bootstrap confidence bands obtained from the traditional smoothing spline (solid) with the true function (dashed). *Lower left*: The curvature of the initial estimate (obtained using $m = 3$) used to weight the smoothing parameter. *Lower middle*: The Loco-spline estimate (solid) with the true function (dashed). *Lower right*: 95% bootstrap confidence bands obtained from the Loco-Spline procedure (solid) with the true function (dashed).

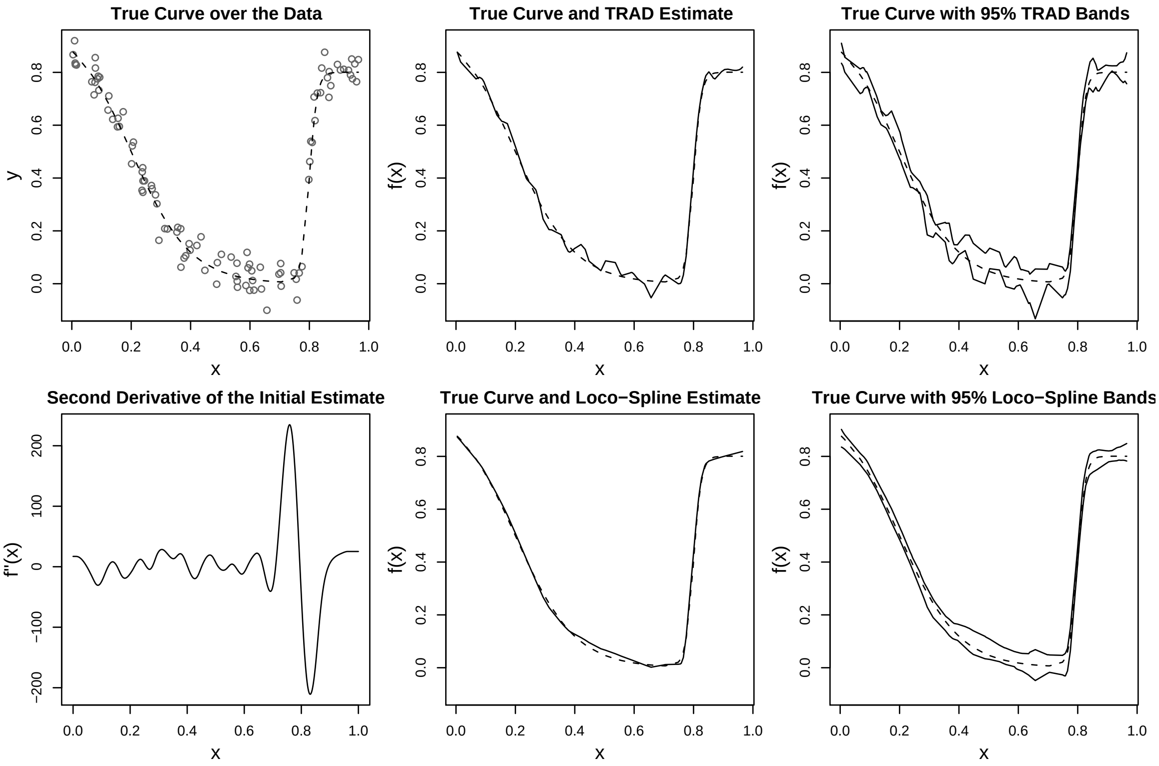

Figure 3: *Upper left*: Data generated from the rapid change function with $n = 100$ along with the true function. *Upper middle*: The traditional smoothing spline estimate (solid) with the true function (dashed). *Upper right*: 95% bootstrap confidence bands obtained from the traditional smoothing spline (solid) with the true function (dashed). *Lower left*: The curvature of the initial estimate (obtained using $m = 3$) used to weight the smoothing parameter. *Lower middle*: The Loco-spline estimate (solid) with the true function (dashed). *Lower right*: 95% bootstrap confidence bands obtained from the Loco-Spline procedure (solid) with the true function (dashed).

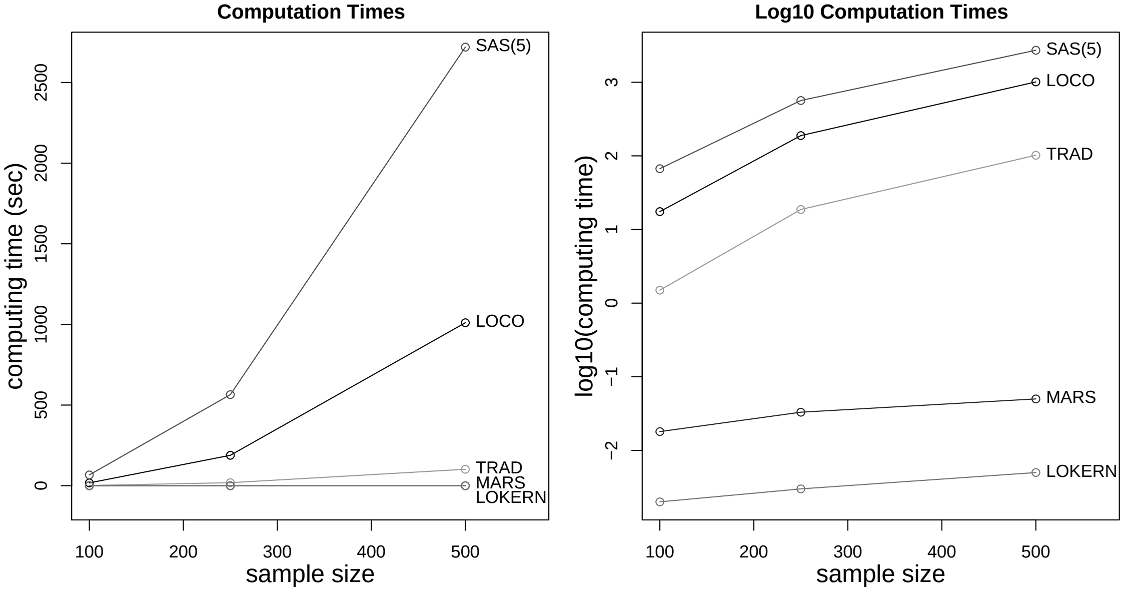

Figure 4: *Left*: Average CPU time (averaged over the simulations from Sections 5.1-5.3) required to fit a given estimator for a given sample size. *Right*: Same as left pane only with Log of Average CPU time.

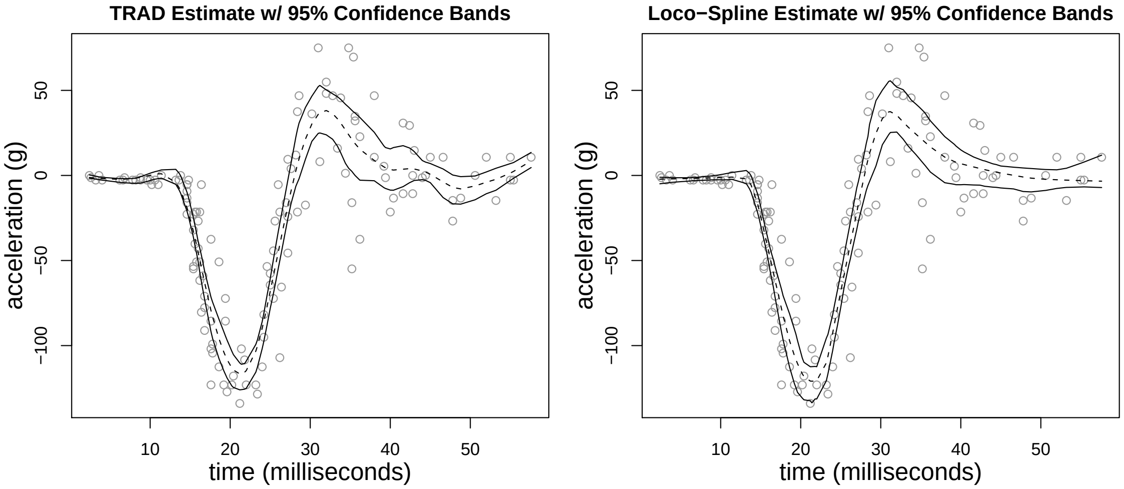

Figure 5: *Left*: Motorcycle crash data along with the estimate given by TRAD (dashed) and 95% bootstrap confidence bands (solid). *Right*: Motorcycle crash data along with the estimate given by Loco-Spline (dashed) and 95% bootstrap confidence bands (solid).

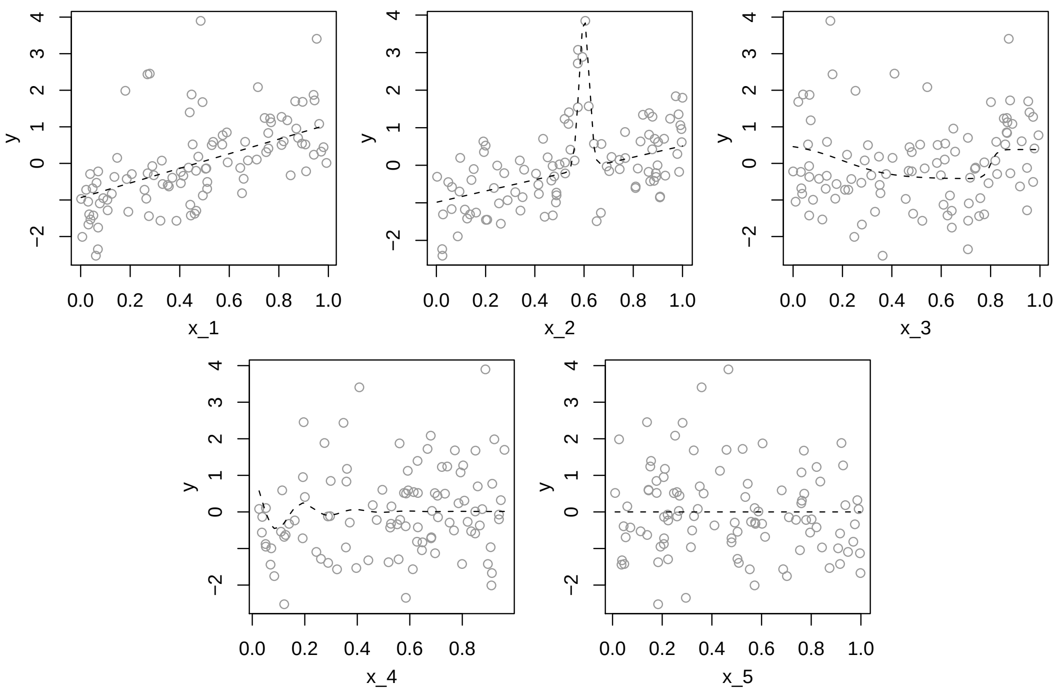

Figure 6: Scatter plots of the data generated from the additive model example across each of the five inputs. The true functional component curves are superimposed.

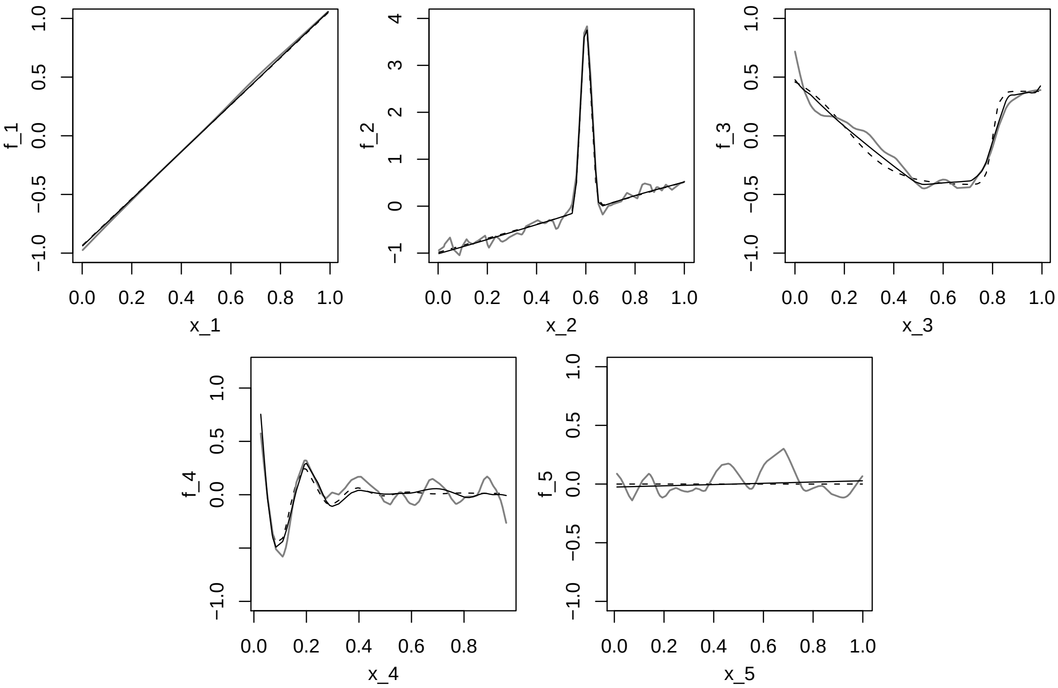

Figure 7: Plot of the true functional component curves (dashed) for the additive model along with the estimates for each component function given by GAM (grey) and the proposed Loco-spline (solid).

Appendix to “A Locally Adaptive Penalty for Estimation of Functions with Varying Roughness” published in the Journal of Computational and Graphical Statistics.

Curtis B. Storlie, Howard D. Bondell, and Brian J. Reich

Date: January 28, 2010

Appendix A: Proofs

The proof of Theorem 1 uses Lemma 1 below which is a generalization of Theorem 10.2 of (van de Geer 2000). Consider the regression model $y_i = g_0(\boldsymbol{x}_i)+\varepsilon_i$, $i = 1, \ldots, n$ where g_0 is known to lie in a class of functions $\mathcal{G}$, $\boldsymbol{x}_i$'s are given covariates in $[0, 1]^p$, and ε_i's are *iid* $\mathcal{N}(0, \sigma^2)$. Let $I_n : \mathcal{G} \to [0, \infty)$ be a pseudonorm on $\mathcal{G}$. Define $\hat{g}_n = \arg\min_{g\in\mathcal{G}} 1/n \sum_{i=1}^{n}(y_i - g(\boldsymbol{x}_i))^2 + \rho_n^2 I_n^v(g)$. Let $H_\infty(\delta, \mathcal{G})$ be the δ-entropy of the function class $\mathcal{G}$ under the supremum norm $\|g\|_\infty = \sup_{\boldsymbol{x}} |g(\boldsymbol{x})|$; see (van de Geer 2000), page 17.

Lemma 1. *Suppose there exists I_* such that $I_*(g) \leq I_n(g)$ for all $g \in \mathcal{G}$, $n \geq 1$. Also assume that there exists constants $A > 0$ and $0 < \alpha < 2$ such that*

$$H_\infty\left(\delta, \left\{\frac{g - g_0}{I_*(g) + I_*(g_0)} : g \in \mathcal{G}, I_*(g) + I_*(g_0) > 0\right\}\right) \leq A\delta^{-\alpha} \qquad \text{(A.1)}$$

for all $\delta > 0$ and $n \geq 1$. Then if $v > 2\alpha/(2 + \alpha)$, $I_*(g_0) > 0$, and $\rho_n^{-1} = O_p\left(n^{1/(2+\alpha)}\right) I_n^{(2v-2\alpha+v\alpha)/(4+2\alpha)}(g_0)$, we have $\|\hat{g}_n - g_0\|^2 = O_p(\rho_n^2) I_n^{v/2}(g_0)$. Moreover, if $I_n(g_0) = 0$ for all $n \geq 1$ then $\|\hat{g}_n - g_0\|^2 = O_p(n^{-v/(2v-2\alpha+v\alpha)})\rho_n^{-2\alpha/(2v-2\alpha+v\alpha)}$.

Proof. This follows the same logic as the proof of Theorem 10.2 of van de Geer (2000), so we have intentionally made the following argument somewhat terse. Notice that

$$\|\hat{g}_n - g_0\|_n^2 + \rho_n^2 I_n^v(\hat{g}_n) \leq 2\left\langle \varepsilon, \hat{g}_n - g_0 \right\rangle_n + \rho_n^2 I_n^v(g_0) \tag{A.2}$$

where $\left\langle \varepsilon, \hat{g}_n - g_0 \right\rangle_n = \sum_{i=1}^n \varepsilon_i \left(\hat{g}_n(x_i) - g_0(x_i)\right)$. Also, condition (A.1) along with Lemma 8.4 in van de Geer guarantees that

$$\sup_{g \in \mathcal{G}} \frac{|\left\langle \varepsilon, \hat{g}_n - g_0 \right\rangle_n|}{\|\hat{g}_n - g_0\|_n^{1-\alpha/2} (I_*(g) + I_*(g_0))^{\alpha/2}} = O_p(n^{-1/2}). \tag{A.3}$$

Case (i) Suppose that $I_*(\hat{g}_n) > I_*(g_0)$. Then by (A.2) and (A.3) we have

$$\begin{aligned} \|\hat{g}_n - g_0\|_n^2 + \rho_n^2 I_n^v(\hat{g}_n) &\leq O_p(n^{-1/2}) \|\hat{g}_n - g_0\|_n^{1-\alpha/2} I_*^{\alpha/2}(\hat{g}_n) + \rho_n^2 I_n^v(g_0) \\ &\leq O_p(n^{-1/2}) \|\hat{g}_n - g_0\|_n^{1-\alpha/2} I_n^{\alpha/2}(\hat{g}_n) + \rho_n^2 I_n^v(g_0). \end{aligned}$$

The rest of the argument is identical to that on page 170 of van de Geer.

Case (ii) Suppose that $I_*(\hat{g}_n) \leq I_*(g_0)$ and $I_*(g_0) > 0$. By (A.2) and (A.3) we have

$$\begin{aligned} \|\hat{g}_n - g_0\|_n^2 &\leq O_p(n^{-1/2}) \|\hat{g}_n - g_0\|_n^{1-\alpha/2} I_*^{\alpha/2}(g_0) + \rho_n^2 I_n^v(g_0) \\ &\leq O_p(n^{-1/2}) \|\hat{g}_n - g_0\|_n^{1-\alpha/2} I_n^{\alpha/2}(g_0) + \rho_n^2 I_n^v(g_0). \end{aligned}$$

The remainder of this case is identical to that on page 170 of van de Geer. □

Proof of Theorem 1. Any function $f(\mathbf{x}) = f_1(x_1) + \cdots + f_p(x_p)$ with each $f_j \in S^m$ can be written as $f(\mathbf{x}) = g_1(\mathbf{x}) + g_2(\mathbf{x})$. The function $g_1(\mathbf{x}) = \alpha_0 + \sum_{j=1}^p \sum_{k=1}^{m-1} \alpha_{jk} x_j^k$ is a parametric additive polynomial part. While $g_2(\mathbf{x}) \in \mathcal{G}$, with

$$\mathcal{G} = \left\{ g_2(\mathbf{x}) = \bar{f}_1(x_1) + \cdots + \bar{f}_p(x_p) : \bar{f}_j \in S^m, \sum_{i=1}^n \bar{f}_j(x_{ij}) x_{il}^k = 0 \text{ for } k = 0, \ldots, m-1 \text{ and } j, l = 1, \ldots, p \right\}.$$

This ensures that $g_2(\mathbf{x})$ is orthogonal to $g_1(\mathbf{x})$ under the empirical dot product, $\langle f, g \rangle = 1/n \sum_{i=1}^n f(\boldsymbol{x}_i) g(\boldsymbol{x}_i)$. Hence $||\hat{f} - f_0||_n^2 = ||\hat{g}_1 - g_{10}||_n^2 + ||\hat{g}_2 - g_{20}||_n^2$. Due to the orthogonality, and that the coefficients on the polynomial terms are unpenalized, it follows that $||\hat{g}_1 - g_{10}||_n^2$ converges with rate n^{-1}.

Now, rewrite the penalty term as $\rho_n^2 \sum_{j=1}^p \int_0^1 \tilde{\lambda}_{j,n}(x) (\bar{f}_j^{(m)}(x))^2 dx$, where $\rho_n^2 = \min\{\lambda_{j,n}(x) : x \in [0,1], j = 1, \ldots, p\}$ and $\tilde{\lambda}_{j,n}(x) = \lambda_{j,n}(x)/\rho_n^2$. The problem is now reduced to showing that the conditions of Lemma 1 hold for the function space $\mathcal{G}$ with $I_n(g) = \left(\sum_{j=1}^p \int_0^1 \tilde{\lambda}_{j,n}(x)(\bar{f}_j^{(m)}(x))^2 dx\right)^{1/2}$, $v = 2$, and ρ_n^2. Notice that by the conditions of Theorem 1 we have $\rho_n^2 \overset{p}{\sim} n^{-2m/(2m+1)}$ and $I_n(g) = O_p(1)$. Also notice that $\tilde{\lambda}_{j,n}(x) \geq 1$ for all $n \geq 1$, $j = 1, ..., p$, and $x \in [0,1]$. This implies that $I_n(g) \geq I_*(g) = \sum_{j=1}^p \int_0^1 (\bar{f}_j^{(m)}(x))^2 dx$ for all $g \in \mathcal{G}$ and $n \geq 1$.

Now the entropy bound in (A.1) holds whenever

$$H_\infty(\delta, \{g \in \mathcal{G} : I_*(g) \leq 1\}) \leq A\delta^{-\alpha}, \tag{A.4}$$

since $I_*(g - g_0) \leq I_*(g) + I_*(g_0)$ so that the set in brackets in (A.4) contains that in (A.1).

Note that $\mathcal{G}$ is a subset of $\bigoplus_{j=1}^p \mathcal{G}_j$, where $\mathcal{G}_j$ is the space for univariate functions after removal of the polynomial in the variable x_j only. Now, for the supremum norm, if for each $\mathcal{G}_j$, subject to $I_*^2(\bar{f}_j) = \int_0^1 (\bar{f}_j^{(m)}(x))^2 dx \leq 1$ can be covered with N balls with radius δ. Then $\bigoplus_{j=1}^p \mathcal{G}_j$, such that $I_*^2(g) \leq 1$ can be covered with N^p balls of size $p\delta$.

Finally, it is known (see for example, van de Geer 2000) that $H_\infty(\delta, \{g \in \mathcal{G}_j : I_*(g) \leq 1\}) \leq A\delta^{-1/m}$. Therefore $H_\infty(p\delta, \{g \in \mathcal{G} : I_*(g) \leq 1\}) \leq Ap\delta^{-1/m}$. So it follows that $H_\infty(\delta, \{g \in \mathcal{G} : I_*(g) \leq 1\}) \leq Ap^{1+\frac{1}{m}}\delta^{-1/m}$. So that the Lemma holds with $\alpha = 1/m$. □

Proof of Corollary 1. For the Loco-spline, we have that $\lambda_{j,n}(x) = \tau_j \left(\left|\tilde{f}_j^{(m)}(x)\right|^{\gamma_j} + \delta_j\right)^{-2}$ with $\tau_j \overset{p}{\sim} n^{-2m/(2m+1)}$. By its construction $\left(\left|\tilde{f}_j^{(m)}(x)\right|^{\gamma_j} + \delta_j\right)^{-2} \overset{p}{\sim} 1$ so the conditions of Theorem 1 are satisfied. □